

# Piecewise-Planar Reconstruction using Two Views

Michel Antunes, João P. Barreto and Urbano Nunes

---

## Abstract

The article describes a reconstruction pipeline that generates piecewise-planar models of man-made environments using two calibrated views. The 3D space is sampled by a set of pre-defined virtual cut planes that pass in between cameras and implicitly define possible pixel correspondences across views. The likelihood of these correspondences being true matches is measured using signal symmetry analysis [1], which enables to obtain profile contours of the 3D scene that become lines whenever the virtual cut planes intersect planar surfaces. The detection and estimation of these *lines cuts* is formulated as a global optimization problem over the symmetry matching cost, and pairs of reconstructed lines are used to generate plane hypotheses that serve as input to PEARL clustering [2]. The PEARL algorithm alternates between a discrete optimization step, which merges planar surface hypotheses and discards detections with poor support, and a continuous optimization step, which refines the plane poses taking into account surface slant. The pipeline outputs an accurate semi-dense PPR of the 3D scene. In addition, the input images can be segmented into piecewise-planar regions using a standard MRF formulation for assigning pixels to plane detections. Extensive experiments with both indoor and outdoor stereo pairs show significant improvements over state-of-the-art methods with respect to accuracy and robustness.

**Keywords:** Piecewise-planar reconstruction, SymStereo, Stereo-Rangefinding, Semi-dense reconstruction

---

## 1. Introduction

Stereo cameras are becoming increasingly popular because of the recent advent of 3D visualization and display. A few years ago they were considered special purpose devices that could only be found in research labs and high-end equipments, but nowadays they are a consumer electronics product being available either as standalone hand-held cameras (e.g. Fujifilm Finepix 3D, Sony Bloggie, etc), or integrated into smart-phones (e.g. HTC Evo 3D). Our work is motivated by this proliferation of stereo cameras that we believe will create an urge for robust algorithms able to render complete, photo-realistic 3D models in an automatic manner.

Stereo reconstruction is a classical problem in computer and robot vision that deserved the attention of thousands of authors [3, 4]. Despite of the many advances in the field, situations of poor texture, variable illumination, severe surface slant or occlusion are still challenging for most stereo matching methods, making it difficult to find a tuning that provides good results under a broad variety of acquisition circumstances [5]. Since man-made environments are dominated by planar surfaces, several authors suggested to overcome the above mentioned difficulties by using the planarity assumption as a prior for the stereo reconstruction [6, 7, 8, 9, 10]. These approaches have the advantage of providing piecewise-planar 3D models of the scene that are perceptually pleasing and geometrically simple, and, thus, their rendering, storage and transmission is computationally less complex. This article proposes a pipeline for two-view *Piecewise-Planar Reconstruction* (PPR) understood as the detection and reconstruction of dominant planar surfaces in the

scene.<sup>1</sup>

PPR is in a large extent a *chicken-and-egg* problem. If there is accurate 3D evidence about the scene, such as points, lines, vanishing directions, etc, then the problem of detecting, segmenting, and estimating the pose of dominant planes can be potentially solved using standard model fitting techniques [13, 2]. On the other hand, if there is a prior knowledge about the dominant planes in the scene, then the matching process can be constrained to improve the accuracy of the final 3D reconstruction, e.g. the known plane orientations can be used to guide the stereo aggregation, as done in [11]. Existing methods for PPR typically comprise three steps that are executed sequentially:

1. *3D Reconstruction:* The objective is to collect 3D evidence about the scene from multiple views. This evidence can either be obtained from *sparse stereo* that matches a sparse set of features across views (e.g. [8, 9]), or from *dense stereo* that performs dense data association between frames by assigning to each image pixel a disparity value (e.g. [10]).
2. *Plane Hypotheses Generation:* Given the 3D data, the objective is to detect and estimate the pose of planar surfaces using some sort of multi-model fitting approach.
3. *Plane Labeling:* The goal is to assign to each image pixel one of the plane hypotheses generated in the previous step. This is usually done using a *Markov Random Field*

---

<sup>1</sup>We mean by PPR something that is different from approximating surfaces by small planes, as typically done in several dense stereo methods (e.g. [11, 12])

(MRF) framework with photo-consistency being used as data term.

While most methods were originally designed to receive multiple views [6, 14, 7, 8, 9, 10], we propose a pipeline that uses only two views and makes no assumptions about the scene other than the fact of being dominated by planar surfaces. The novelty is mainly in the steps of *3D Reconstruction* and *Plane Hypothesis Generation*, and the contributions can be summarized as follows:

- *Reconstruction of line cuts using Stereo from Induced Symmetry (SymStereo)*: Establishing dense stereo correspondence is computationally expensive specially when dealing with high-resolution images. On the other hand, sparse stereo applied to only two views tends to provide insufficient 3D data for establishing accurate plane hypotheses. Thus, we propose to carry a semi-dense reconstruction of the scene by independently recovering depth along a set of pre-defined virtual planes using SymStereo [1]. This approach is known as *Stereo-Rangefinding*, because the result are profile cuts of the scene similar to the ones that would be obtained using a Laser-Rangefinder [15]. Since the intersection of the virtual scan planes with planes contained in the scene are lines, we extract line segments from the profile cuts and use these *line cuts* to generate plane hypotheses [16].
- *Improving SymStereo accuracy in the case of surface slant*: In a similar manner to what happens in conventional stereo, surface slant affects the depth estimation obtained from SymStereo. In this case, the line cuts are poorly reconstructed, which also implies that the detected planes are inaccurate. We study the problem of surface slant in the context of the SymStereo framework, and devise a simple solution that enables an accurate reconstruction of highly slanted planes.
- *Global plane fitting using PEARL [2]*: Most methods for PPR treat stereo matching and plane detection in a sequential and independent manner [6, 14, 7, 8, 9, 10]. This is problematic because the accuracy of the plane hypotheses is inevitably limited by the accuracy of the initial 3D reconstruction that does not take into account the fact of the scene being dominated by planar surfaces. We carry the 3D reconstruction and the plane fitting in a simultaneous and integrated manner using the recent PEARL framework proposed in [2]. The algorithm alternates between a global discrete optimization step, which merges plane surface hypotheses and discards spurious detections, and a continuous optimization step over the symmetry energy, which refines the plane pose estimation taking into account surface slant. The output is a set of plane hypotheses and a semi-dense PPR of the 3D scene, where the reconstructed line cuts are labeled according to the plane detections.

### 1.1. Related Work

Several works in PPR start by obtaining a sparse 3D reconstruction of the scene (e.g. point clouds, edge lines, etc), then

establish plane hypotheses by applying multi-model fitting to the reconstructed data, and finally use these hypotheses to guide the dense stereo process and/or perform a piecewise-planar segmentation of the input images [6, 14, 7]. Werner and Zisserman use multiple cues and assumptions to find dominant surface orientations, and then perform plane-sweep reconstruction along the detected normal directions. Pollefeys et al [7] detect planar surfaces in urban environments from 3D point features obtained from SfM, and use the estimated normals for guiding plane-sweep stereo.

Furukawa et al [8] propose to perform PPR assuming a Manhattan-world model. They reconstruct 3D patches in textured image regions from multiple views using [17], and use the normals of these patches to establish plane hypotheses. These hypotheses are then used in a MRF formulation for pixel-wise plane labeling. In [9], Sinha et al. introduce a probabilistic framework for assigning plane hypotheses to pixels with the evidences of planar surfaces being provided by point cloud reconstruction, estimation of vanishing lines, and sparse reconstruction of edges. Gallup et al [10] propose a stereo method capable of handling both planar and non-planar objects contained in the scene. A robust procedure based on RANSAC is used for fitting plane hypotheses to dense depth maps, followed by a MRF formulation for plane labeling of the input images.

An alternative strategy is to over-segment the stereo images based on color information and fit a 3D plane to each non-overlapping region. The number of planes to be considered is defined by the segmentation result, which acts as a smoothness prior during the global optimization. This segmentation information is either used as a hard minimization constraint [18, 19, 20] or as a soft constraint [21]. The main weakness of this type of strategy is the assumption that planar surfaces in the scene have different colors, which is often not the case in most man-made environments (e.g. walls, doors, windows, etc).

The drawback of the approaches described so far is the fact that depth estimation and plane fitting are carried in a sequential and decoupled manner. The errors in the extracted 3D evidence may affect the accuracy of the plane pose estimation, and the inferred planar surfaces are not used for refining the initial depth estimates.

There are a few approaches [22, 23, 24] that perform PPR by carrying stereo matching and 3D plane fitting iteratively. The strategy consists in alternating between segmenting the input images into non-overlapping regions and estimating the plane parameters for each region. However, and as stated by the authors of [23], these types of algorithms can become easily stuck in a local minimum whenever they face challenging surface structures e.g. surfaces with low and/or repetitive texture.

### 1.2. Article Overview and Notation

Section 2 reviews three background concepts that are used throughout the article. Section 3 proposes an algorithms for reconstructing line cuts along a single virtual cut plane, and Section 4 shows how the SymStereo estimates can be refined in case there is prior slant information available. Then, Section 5



present an algorithm for semi-dense PPR that combines SymStereo and PEARL. Finally, Section 6 reports experiments in PPR, where the accuracy of the plane estimation and pixel labeling is evaluated with respect to ground truth data, and the performance of our pipeline is compared with two different strategies. The results show that the plane hypotheses computed using our symmetry-based algorithm outperforms the approaches based on dense stereo reconstruction and sparse feature matching.

We represent scalars in italic, e.g.  $s$ , vectors in bold characters, e.g.  $\mathbf{p}$ , matrices in sans serif font, e.g.  $\mathbf{M}$ , and image signals in typewriter font, e.g.  $\mathbf{I}$ . Unless stated otherwise, we use homogeneous coordinates for points and other geometric entities, e.g. a point with non-homogeneous image coordinates  $(p_1, p_2)$  is represented by  $\mathbf{p} \sim (p_1 \ p_2 \ 1)^T$ , with  $\sim$  denoting equality up to scale.

## 2. Background

This section briefly reviews three background concepts that are used throughout the article, namely Stereo-Rangefinding using SymStereo (Section 2.1), the energy-based multi-model fitting framework called PEARL (Section 2.2), and a global pixel-wise plane labeling formulation (Section 2.3). There is no major novelty, so that readers that are familiar with these concepts can skip this section.

### 2.1. Stereo-Rangefinding using SymStereo

The SymStereo framework [1] was proposed for matching pixels across stereo views using symmetry analysis instead of traditional photo-consistency. Let  $\mathbf{I}$  and  $\mathbf{I}'$  be a pair of rectified stereo images and consider a virtual cut plane  $\Pi$  (see Figure 1). The orientation of the virtual plane is arbitrary being the only requirement that it intersects the baseline. Under such circumstances, the left and right back-projections become reflected one with respect to the other at the locations where the virtual plane intersects the scene. Thus, the sum of both back-projections gives rise to an image signal that is locally symmetric around the *profile cut*, while the subtraction results in a signal that is anti-symmetric. These symmetries are usually not *strict symmetries* due to perspective distortion, surface slant and occlusions, but can be used as cues to recover the profile cut where the virtual plane meets the scene.

Assuming that the world coordinate system is coincident with the reference frame of the left view, the virtual cut plane  $\Pi$  can be represented by the homogeneous vector

$$\Pi \sim (\mathbf{n} \ -h)^T, \quad (1)$$

where  $\mathbf{n}$  indicates the direction orthogonal to the plane

$$\mathbf{n} \sim (n_1 \ n_2 \ n_3)^T.$$

The homogenous coordinates of the intersection point  $\mathbf{O}$  of the virtual cut plane with the baseline is given by [1]:

$$\mathbf{O} \sim \left( \frac{h}{n_1} \ 0 \ 0 \ 1 \right)^T.$$

Using  $\beta$  to denote the ratio between  $O_1$  and the baseline length  $b$  comes that the plane  $\Pi$  passes between the cameras *iff* the following condition holds

$$0 < \beta < 1.$$

For efficiency purposes, the images do not need to be explicitly back-projected onto the virtual plane  $\Pi$ , but instead the homography  $\mathbf{H}$  induced by  $\Pi$  can be used to map points from the right view into the left view [1]:

$$\mathbf{H} \sim \begin{pmatrix} 1 + \frac{bn_1}{h-bn_1} & \frac{bn_2}{h-bn_1} & \frac{bn_3}{h-bn_1} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Assuming that  $\hat{\mathbf{I}}$  is the warping result of mapping  $\mathbf{I}'$  using  $\mathbf{H}$ , then it comes from the mirroring effect [1] that  $\mathbf{I}$  and  $\hat{\mathbf{I}}$  are reflected around the image of the profile cut (see Figure 1). Thus, the sum of  $\mathbf{I}$  and  $\hat{\mathbf{I}}$  yields an image signal  $\mathbf{I}^S$  that is symmetric around the locus where the profile cut is projected. In a similar manner, the difference between  $\mathbf{I}$  and  $\hat{\mathbf{I}}$  gives rise to an image signal  $\mathbf{I}^A$  that is anti-symmetric at the exact same location. SymStereo detects the image of the profile cut by jointly evaluating the symmetry and anti-symmetry in  $\mathbf{I}^S$  and  $\mathbf{I}^A$ . This provides an implicit manner of recovering depth along  $\Pi$  and achieving data association across views.

Since the symmetries are induced using virtual cut planes, SymStereo is particularly well suited for recovering depth along pre-defined scan planes. As discussed in [1], this is an effective way of probing into the 3D structure, resulting in profile cuts that resemble the ones obtained with a Laser-Rangefinder [15]. The independent estimation of depth along a scan plane is called Stereo-Rangefinding. It was concluded in [1] that the  $\log N$  matching cost is the top-performer metric for this purpose. The  $\log N$  cost is based on local frequency analysis for locating symmetric structures by employing a bank of  $N$  log-Gabor wavelets (we set  $N = 10$  in this article). The output of  $\log_{10}$  is the joint energy  $E$ , where the image of the profile cut is highlighted (see Figure 1).

### 2.2. Energy-based multi-model fitting using PEARL

Isack and Boykov argued in [2] that methods that greedily search for models with most inliers while ignoring the overall classification of data are a flawed approach to multi-model fitting, and that formulating the fitting as an optimal labeling problem with a global energy function is preferable. For this purpose, they propose the PEARL algorithm consisting in three main steps:

1. *Propose* an initial set of plausible models (labels)  $\mathcal{L}_0$  from the observations
2. *Expand* the label set for estimating its spatial support (inlier classification)
3. *Re-estimate* the inlier models by minimizing some error function.

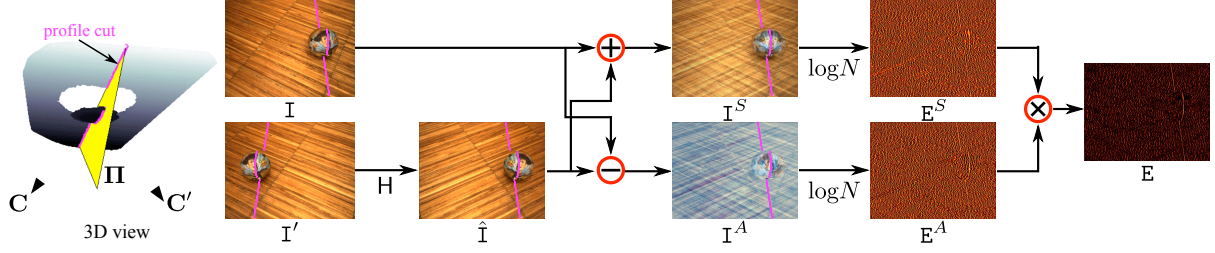


Figure 1: The virtual cut plane  $\Pi$  (yellow) passes between the cameras and intersects the 3D scene in a non-continuous 3D curve (magenta). Let  $\hat{I}$  be the result of warping  $I'$  by the homography induced by  $\Pi$ . The images  $I^S$  and  $I^A$  are, respectively, symmetric and anti-symmetric around the image of the profile cut (magenta). The output of the  $\log N$  joint symmetry and anti-symmetry quantification method is the energy map  $E$  that highlights the image of the profile cut.

Given the initial model set  $\mathcal{L}_0$ , the multi-model fitting is cast as a global optimization where each model in  $\mathcal{L}_0$  is interpreted as a particular label  $f$ . Consider that  $d \in \mathcal{D}$  is a data point and that  $f_d$  is a particular label in  $\mathcal{L}_0$  assigned to  $d$ . The objective is to compute the labeling  $\mathbf{f} = \{f_d | d \in \mathcal{D}\}$  such that the following energy is minimized:

$$E(\mathbf{f}) = \underbrace{\sum_{d \in \mathcal{D}} D_d(f_d)}_{\text{data term}} + \underbrace{\lambda_S \sum_{d, e \in \mathcal{N}} V_{d,e}(f_d, f_e)}_{\text{smoothness term}} + \underbrace{\lambda_L \cdot |\mathcal{F}_f|}_{\text{label term}} \quad (3)$$

where  $\mathcal{N}$  is the neighborhood system considered for  $d$ ,  $D_d(f_d)$  is some error that measures the likelihood of point  $d$  belonging to model  $f_d$ , and  $V_{d,e}$  is the spatial smoothness term that encourages piecewise smooth labeling by penalizing configurations  $\mathbf{f}$  that assign to neighboring nodes  $d$  and  $e$  different labels. The label term is used for describing the data points using as few unique models as possible, with  $\mathcal{F}_f$  being the subset of different models assigned to the nodes  $d$  by the labeling  $\mathbf{f}$  (see [2] for further details). In order to handle outlier data points in  $\mathcal{D}$ , the outlier label  $f_\emptyset$  is added to  $\mathcal{L}_0$ . Any point  $d$  to which is assigned the label  $f_\emptyset$  is considered an outlier, and has usually a constant likelihood measure  $D_d(f_d = f_\emptyset) = \tau$ . The energy of Equation 3 is efficiently minimized using  $\alpha$ -expansion [2].

Finally, the third step of PEARL consists in re-estimating the model labels  $f$  in  $\mathcal{L}_0$  with non-empty set of inliers  $\mathbf{D}(f) = \{d \in \mathcal{D} | f_d = f\}$ . Let  $\mathbf{m}_f$  be the model associated to the label  $f$ . Each model  $\mathbf{m}_f$  is refined by minimizing the error cost over its parameters:

$$\mathbf{m}_f^* = \min_{\mathbf{m}_f} \sum_{d \in \mathbf{D}(f)} D_d(f).$$

The models with non-empty set in  $\mathcal{L}_0$  are replaced with the refined models  $\mathbf{m}_f^*$ , and the labels with empty set are discarded. The new set of labels  $\mathcal{L}_1$  is then used in a new expand step, and we iterate between discrete labeling and plane refinement until the  $\alpha$ -expansion optimization does not decrease the energy of Equation 3.

### 2.3. MRF for Plane Labeling

Given a set of plane hypotheses in the scene, the following step of most PPR algorithms is to compute a pixel-wise plane labeling of the input images. We follow a standard MRF formulation for comparing all the tested algorithms. The objective is

to minimize an energy involving data, smoothness and labeling terms (refer to Equation 3). In this case, the nodes  $d \in \mathcal{D}$  are the image pixels, and the labels  $f \in \mathcal{P}$  are the plane hypotheses. A  $4 \times 4$  neighborhood  $\mathcal{N}_4$  is assumed for neighboring pixels  $d$  and  $e$ , and the data term is defined as

$$D_d(f) = \begin{cases} \min(\rho_d(f), \rho_{max}) & \text{if } f \in \mathcal{P} \\ \gamma \rho_{max} & \text{if } f = f_\emptyset \end{cases} \quad (4)$$

where  $\rho_d(f)$  is the photo-consistency between the pixels in the two views put into correspondence by the plane associated to label  $f$ . For measuring the photo-consistency, we use Zero-mean Normalized Cross-correlation (ZNCC). The photo-consistency metric is given by  $\rho_d(f) = (1 - ZNCC(f))/2$ , where  $ZNCC(f)$  is the cost obtained using ZNCC for the plane hypothesis  $f$  ( $\rho_{max}$  and  $\gamma$  are constant parameters).

The smoothness term is defined as:

$$V_{d,e}(f_d, f_e) = g \cdot \begin{cases} 0 & \text{if } f_d = f_e \\ M & \text{if } (f_d \vee f_e) = f_\emptyset \\ D' & \text{otherwise} \end{cases} \quad (5)$$

where

$$D' = \min(D, M) + m \quad \text{and} \\ g = \frac{1}{\Delta I^2 + 1};$$

$D$  is the 3D distance between neighboring points according to their plane  $f_d$  and  $f_e$ , respectively,  $M$  and  $m$  are constant parameters, and

$$\Delta I = |I(d) - I(e)|.$$

is the image gradient.

The data and the smoothness terms are, with minor differences, similar to the ones used in the graph-cut labeling of Gallup et al. [10]. We additionally add the labeling term  $\lambda_L |\mathcal{F}_f|$  for avoiding very close plane hypotheses in  $\mathcal{P}$  to be assigned in  $\mathbf{f}$ . This has the effect of simplifying the 3D model using as few unique planes as possible.

## 3. Reconstruction of Lines along a single cut plane

The reconstruction of lines from two or more views has in the vast majority of existing algorithms one common denominator: the detection of line segments in the input views that are

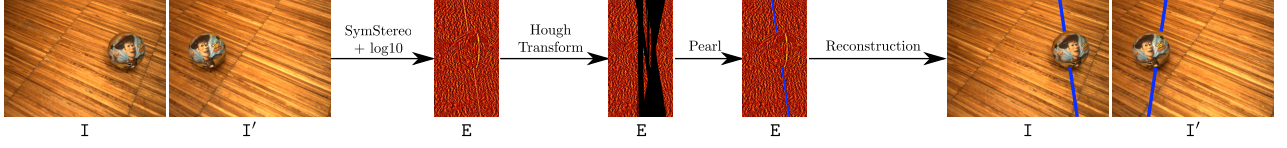


Figure 2: Reconstruction of 3D line cuts from a stereo pair along a virtual cut plane  $\Pi$ . We use SymStereo and employ the  $\log_{10}$  symmetry-based matching cost for obtaining the joint energy  $E$ . The energy  $E$  is used as input to a weighted Hough transform for extracting line cuts (black lines), from which the most appropriate hypotheses (in this example only one line cut (blue) is detected) are selected using a global framework constituted by data, smoothness and label costs.

matched in subsequent steps. In the case there are no (salient) line segments in the input images, then no 3D line reconstructions can be obtained.

This section describes an algorithm that reconstructs a set of 3D line cuts along a single virtual cut plane  $\Pi$  using Stereo-Rangefinding. This is achieved by noting that the intersection of  $\Pi$  with a plane in the scene is a line (e.g. in Figure 2, the intersection of  $\Pi$  with the floor plane is the blue line cut). The 3D lines corresponding to the intersection of  $\Pi$  with multiple planes are projected onto the stereo views as line segments, whose locations in most of the cases cannot be perceived only from the input images alone (there are no visible edges). However, these lines can be reliably detected and estimated from the joint symmetry and anti-symmetry energy  $E$  that is obtained from  $\Pi$ . Remark that each line cut that is detected from a virtual cut plane corresponds to a particular plane contained in the scene. However, the corresponding parameters cannot be estimated from a single cut plane (we will see in Section 5 how to detect and estimate planes based on the information of more than one virtual cut plane).

### 3.1. Line cut detection using Hough and PEARL

As shown in Figure 2, we use the SymStereo framework along a virtual cut plane  $\Pi$  and employ the  $\log_{10}$  symmetry metric for computing the joint energy  $E$ . Each pixel in  $E$  provides the matching likelihood of a particular pair of pixels in the stereo views, being an indirect measurement of the occupancy probability in 3D along  $\Pi$ . The energy  $E$  is used as input to a weighted Hough transform for extracting a set of line cut hypotheses  $\mathcal{L}_0$ . This is accomplished by selecting the  $N_H$  local maxima in the Hough voting space.

Next, we formulate the line cut detection as a global labeling problem in a PEARL framework, in which the objective is to assign to each epipolar line (image row) a line cut hypothesis in  $\mathcal{L}_0$ . Following the notation of Section 2.2, the data points  $d$  of the graph are the epipolar lines, with the size of the set  $\mathcal{D}$  being equal to the number of image rows, and the goal is to assign a line segment label  $f$  to each epipolar line  $d$ . The data term is defined as

$$D_d(f) = \begin{cases} \min(1 - E(d, x_f), \tau) & \text{if } f \neq f_\emptyset \\ \alpha_\emptyset \tau & \text{otherwise} \end{cases}$$

where  $E(r, c)$  is the joint energy value for row  $r$  and column  $c$ . The coordinate  $x_f$  corresponds to the intersection between the epipolar line  $d$  and the line segment  $\mathbf{l}_f$  associated to label  $f$ . Remark that the truncation parameter  $\tau$  is used for handling poorly matching surfaces e.g. containing low and/or repetitive

textures, while the discard label  $f_\emptyset$  indicates that no satisfactory line cut hypothesis can be assigned to  $d$ . In this case, the virtual cut plane  $\Pi$  has high probability of not intersecting a planar surface along the epipolar plane associated to  $d$ .

The smoothness term of neighboring nodes  $d$  and  $e$  is given by

$$V_{de}(f_d, f_e) = \begin{cases} 0 & \text{if } f_d = f_e \\ \lambda_\emptyset & \text{if } (f_d \vee f_e) = f_\emptyset \\ \frac{1}{\Delta I^2 + 1} & \text{otherwise} \end{cases}$$

where

$$\Delta I = |I(d, x_{f_d}) - I(e, x_{f_e})|$$

is the image gray-scale gradient. No penalization is assigned to neighboring image rows  $d$  and  $e$  receiving the same label, while in the case one node receives the label  $f_\emptyset$ , then a non-zero cost  $\lambda_\emptyset$  is added to  $\mathbf{f}$ . The smoothness term  $V$  prefers label transitions at locations of larger image gradient (lower smoothness cost), which usually occurs at the boundaries of two different surfaces. We use a constant label term  $\lambda_L$  in Equation 3 for favoring line cut assignments  $\mathbf{f}$  with fewer labels.

Finally, and after computing an initial labeling solution  $\mathbf{f}$  for nodes  $d$ , the line cuts  $\mathbf{l}$  are refined by minimizing their parameters over the energies  $E$  via Levenberg-Marquardt (LM) [25]

$$\mathbf{l}_f^* = \min_{\mathbf{l}_f} \sum_{d \in \mathbf{D}(f)} (1 - E(d, x_f)), \quad (6)$$

where  $\mathbf{D}(f)$  is a subset of image rows  $d$  to which the label  $f$  was assigned. Remark that at each solver iteration, the point  $x_f$  on  $d$  is recomputed according to the current line cut hypothesis  $\mathbf{l}_f$ . The new set of line cuts  $\mathbf{l}_f^*$  are then used in a new global line cut assignment (expand) step, and we iterate between discrete labeling and line cut refinement until the energy of Equation 3 stops decreasing (which usually occurs after 2–3 iterations).

### 3.2. Experiments in line cut detection

We performed experiments of our line cut detection approach<sup>2</sup> on various indoor scenes (see Figure 2-4) acquired using a Bumblebee stereo camera from PointGrey, which has a baseline of 24 cm and image resolution of  $1024 \times 768$  pixels.

In the first example of Figure 3, we detect 2 different line cuts for each virtual cut plane, one corresponds to the intersection of the cut planes with the floor and one is due to the

<sup>2</sup>We used for all the experiments the same parameters:  $N_H = 200$ ,  $\lambda_S = 1$ ,  $\tau = 0.8$ ,  $\alpha_\emptyset = 0.7$ ,  $\lambda_\emptyset = 0.9$  and  $\lambda_L = 20$ , which were empirically selected without much effort.

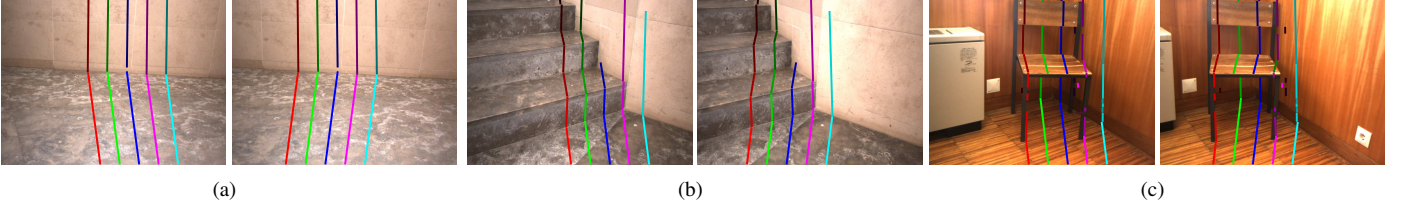


Figure 3: Results produced by our line cut detection algorithm along 5 virtual cut planes. We show for each example the left and right views with the detected line cuts overlaid (compare the matching between the views), different colors indicate different cut planes, while different shades identifying different line cuts.

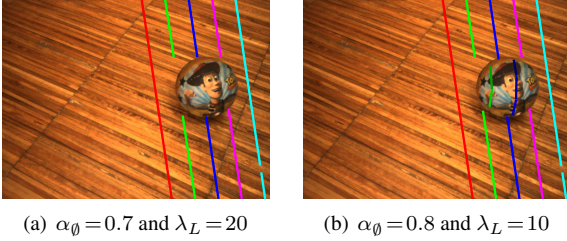


Figure 4: Results for two different settings of  $\alpha_\theta$  and  $\lambda_L$ . By varying these parameters, we can control the algorithm to be more permissive with respect to what is considered a line cut (b), while for lower values of  $\alpha_\theta$  and higher values of  $\lambda_L$  the algorithm only detects line segments with high probability of belonging to planar surfaces (a).

intersection with the wall. Remark that the matching of the line segments across the views is almost perfect and consistent for all virtual cut planes, even though the line cut detection was carried for each virtual cut plane independently. In the example (b), the scene consists of multiple planar surfaces, some containing quite complicated textures. In this case, the line cut estimation approach along a single virtual cut plane begins to have difficulties. In situations the cut plane intersects the scene in low-textured regions, the symmetry based matching using  $\log_{10}$  does not provide a well defined ridge at the locations of the image of the profile cut. Following this, the algorithm prefers to label those regions with the  $f_\theta$  label (e.g. blue cut plane), since it has low confidence about the location of the image of the profile cut. Finally, the example (c) presents some failure cases of this approach, namely slanted surfaces with low-texture. In this cases, the algorithm tends to (i) assign more than one line cut label that corresponds to the same planar surface (noisy energy E), (ii) does not detect the line cut at all, or (iii) computes wrong matches. Note that (i) could be handled by increasing the label cost  $\lambda_L$ , however this would imply that line cuts corresponding to close planes (e.g. chair backs and wall in example (c)) are assigned the same label. We will show in Section 4 that these estimations can be considerably refined, and in Section 5 that most of the difficulties are easily handled by our PPR algorithm that jointly estimates plane hypotheses from multiple virtual cut planes simultaneously.

So far most of the examples contained only planar surfaces. We show in Figure 4 a scene containing a non-planar object above the floor plane. The control of labeling just *strict* planes (example (a)) or approximate non-planar surfaces by an appropriate set of planes (example (b)) is achieved using different settings of the weighting factor  $\alpha_\theta$  and the label cost  $\lambda_L$ . Us-

ing low values of  $\alpha_\theta$  and high values of  $\lambda_L$  implies that only line cuts belonging to planar surfaces are reconstructed, while higher values of  $\alpha_\theta$  and low values of  $\lambda_L$  enable to approximate non-planar surfaces by various plausible line cuts.

#### 4. Line cut detection under surface slant

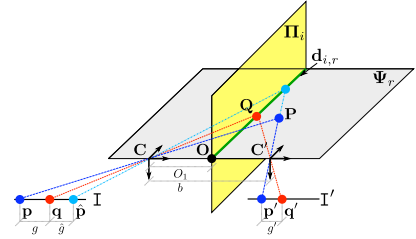


Figure 5: Geometric overview of SymStereo. The camera centers  $C$  and  $C'$  are separated by a distance  $b$  (the baseline). The 3D point  $Q$  on  $d_{i,r}$  is detected using the mirroring effect induced by the virtual cut plane  $\Pi_i$  (yellow) intersecting the baseline.

The SymStereo framework was introduced in [1], where a thorough geometric analysis was provided. This section extends [1], and studies the problem of surface slant in the context of SymStereo. In plane-sweeping [26] it is possible to integrate prior knowledge of the scene to select the sweeping directions that maximize the performance of photo-consistency based stereo (e.g. [27, 6]). We show that slant priors can also be used in SymStereo for choosing the cut planes that render perfect signal symmetries, and improve the overall accuracy and robustness of the approach.

Consider a generic point  $P$  and a point  $Q$  that lies on the same epipolar plane  $\Psi$ , and also assume that  $Q$  belongs to the virtual cut plane  $\Pi$  (see Figure 5). Let  $p$  and  $p'$ , and  $q$  and  $q'$  be the image projections of  $P$  and  $Q$ , respectively. The signed distances between the images of the two points are defined as:

$$g = p_1 - q_1, \quad g' = p'_1 - q'_1. \quad (7)$$

It is shown in [1] that:

$$\hat{g} = \hat{p}_1 - q_1 = \left( \frac{\beta}{\beta - 1} \right) g', \quad (8)$$

where  $\hat{p} \sim H p'$  (refer to Equation 2). Let  $\Delta_p = p_1 - p'_1$  and  $\Delta_q = q_1 - q'_1$  be the stereo disparities of  $P$  and  $Q$ , respectively, and define

$$\Delta = \Delta_p - \Delta_q.$$



From Equation 7 follows that  $g' = g - \Delta$ , and Equation 8 can be written as

$$\hat{g} = \left( \frac{\beta}{\beta - 1} \right) (g - \Delta). \quad (9)$$

The deviation in perfect mirroring ( $\hat{g} = -g$ ) around the projection of the profile cut is function of the differences in pixel disparities, which is directly related to the depth variation in the neighborhood of the 3D profile cut. Note that the virtual cut plane  $\Pi$  only affects the symmetry in terms of the intersection point with the baseline. For similar conditions of relative depth variation, any cut plane going through the same point  $O$  generates symmetries with equivalent quality, regardless of its orientation. Also note that, for the particular case of planes  $\Pi$  intersecting the baseline in the midpoint ( $\beta = 0.5$ ), the symmetry is perfect whenever the surfaces to reconstruct are fronto-parallel to the stereo rig ( $\Delta = 0$ ).

#### 4.1. Slant prior for enhancing SymStereo

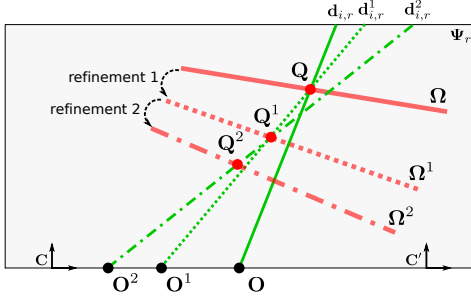


Figure 6: Refinement using slant prior (top view of scene in Figure 5). Assume that  $Q$  lies on the plane  $\Omega$ . Then, we can determine the position on the baseline  $O^1$  (see Equation 12) that improves the induced symmetries. Using the vertical virtual cut plane defined by  $O^1$  and  $Q$ , it is possible to induce new symmetries from which the refined point  $Q^1$  is estimated.

Assume that the points  $P$  and  $Q$  also lie on the same scene plane  $\Omega \sim (\mathbf{m} \ -l)^T$  that defines a homography  $M$ , similar to Equation 2, mapping points in the right view into points in the left view. Following this,  $\mathbf{q} = M\mathbf{q}'$  and it can be shown that

$$\Delta_q = \frac{m_1 b}{l} q_1 + \frac{m_2 b}{l} q_2 + \frac{-m_1 b q_1 - m_2 b q_2 + l \Delta_q}{l}.$$

Since  $\mathbf{p}$  is also the projection of the same planar surface, by applying the homography  $M$  comes that  $\Delta_p$  differs from  $\Delta_q$  by

$$\Delta = \alpha_1 (p_1 - q_1).$$

where

$$\alpha_1 = \frac{m_1 b}{l} \quad (10)$$

is proportional to the slant of the plane along the horizontal direction. Replacing in Equation 9 comes that

$$\hat{g} = \left( \frac{\beta}{\beta - 1} \right) (1 - \alpha_1) g. \quad (11)$$

The conclusion that can be drawn is that having prior knowledge about the position and orientation of the surface to be reconstructed, we can determine the point of intersection between

the virtual plane  $\Pi$  and the baseline that grants perfect induced symmetry. The image signals are perfectly symmetric whenever  $\hat{g} = -g$ , so that solving with respect to  $\beta$  in Equation 11 yields

$$\beta = \frac{1}{2 - \alpha_1}. \quad (12)$$

Following the previous analyze, and in case there is slant information available a priori, we suggest a simple approach for refining the SymStereo depth estimates. Referring to Figure 6, we start by applying a virtual cut plane  $\Pi_i$  intersecting the baseline in its midpoint  $O_1 = 0.5b$ , from which the 3D point  $Q$  is estimated. Assume that  $Q$  lies on the plane  $\Omega$ , whose horizontal slant defines a particular direction  $\alpha_1$  (Equation 10). Using Equation 12, we can determine the position on the baseline  $O_1^1 = \beta^1 b$  that a new virtual cut plane should intersect for enhancing the quality of the induced symmetries. This new vertical virtual cut plane  $\Pi_i^1$  is defined by the points  $O_1^1$  and  $Q$ , from which a refined 3D estimation  $Q^1$  can be computed. Following this, the overall quality of the 3D points obtained using SymStereo can be iteratively refined by selecting appropriate virtual planes intersecting specific points on the baseline.

## 5. PPR using SymStereo and PEARL

This section describes an algorithm that combines SymStereo (Section 2.1) with the geometric multi-model fitting algorithm PEARL (Section 2.2) for semi-dense PPR (see Figure 7). The input to the algorithm are  $M$  joint energies  $E_i$  that were computed using  $\log_{10}$  from a set of  $M$  virtual cut planes  $\Pi$  that belong to a vertical pencil intersecting the baseline in its midpoint. The output are a discrete set of planar surfaces and a semi-dense 3D reconstruction, where each reconstructed point belongs to a particular plane. The detected planes can then be used as plane hypotheses in a global plane labeling strategy for computing a dense model (see Section 2.3).

### 5.1. Formulation of the global framework

Consider that the midpoint of the baseline is the center of projection of a virtual camera, which will be called the cyclopean eye (see Figure 8). The image that is perceived by the cyclopean eye has height equal to the number of epipolar planes  $\Psi_r$  with  $r = 1, \dots, R$  (one epipolar plane per image row), and the width is given by number of virtual cut planes  $\Pi_i$  with  $i = 1, \dots, M$  (one cut plane for each column). Each pixel of the cyclopean eye is originated by the back-projection ray  $\mathbf{d}_{i,r}$ , corresponding to the intersection between  $\Pi_i$  and  $\Psi_r$ . The objective is to estimate the point on each  $\mathbf{d}_{i,r}$  that most likely belongs to a planar surface. This problem is cast as a labeling problem following a PEARL framework (Section 2.2). The nodes of the graph are the back-projection rays  $\mathbf{d}_{i,r}$  of the cyclopean eye, and to each  $\mathbf{d}_{i,r}$  we want to assign a plane label  $f_d$ . The set of possible labels is  $\mathcal{L}_0 = \{\mathcal{P}_0, f_\emptyset\}$ , with  $f_\emptyset$  meaning that no point on  $\mathbf{d}_{i,r}$  belongs to a planar surface. Note that we use  $\mathbf{d}$  instead of  $\mathbf{d}_{i,r}$  whenever the virtual and epipolar plane specifications are not strictly necessary. We assume a  $\mathcal{N}_4$  neighborhood for  $\mathbf{d}_{i,r}$  that is defined by the four back-projection rays  $\mathbf{d}_{i\pm 1,r}$  and  $\mathbf{d}_{i,r\pm 1}$  (see Figure 8).

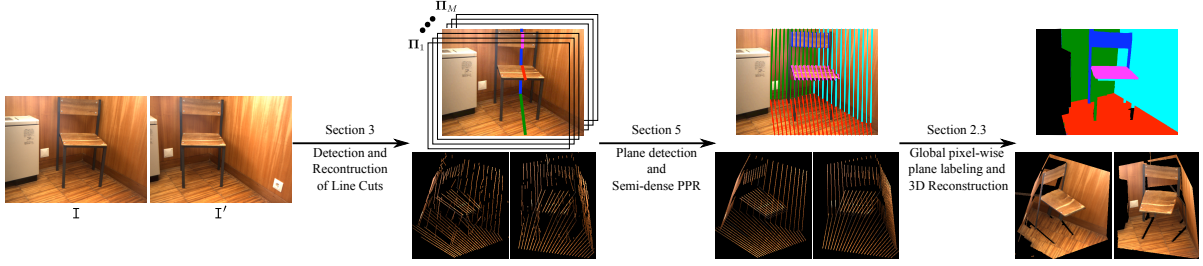


Figure 7: Pipeline for PPR using a pair of calibrated images: (1) apply the line cut detection algorithm described in Section 3 along  $M$  virtual cut planes for obtaining a sparse set of 3D line cuts; then (2) use the global semi-dense PPR algorithm described in Section 5 for computing planar surfaces and obtain a semi-dense PPR; use the line cuts estimated in (1) for obtaining plane hypotheses; and (3) use the global pixel-wise plane labeling (Section 2.3) for computing a dense PPR model from the plane hypotheses in (2).

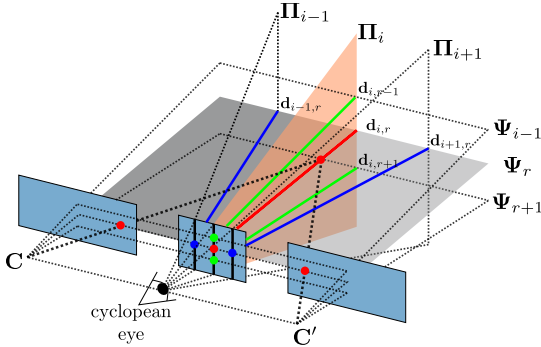


Figure 8: The scene is sampled by a discrete set of virtual cut planes  $\Pi_i$ . This can be thought as an image created by a virtual camera that is located between the cameras (cyclopean eye), where each epipolar plane  $\Psi_r$  projects onto one row and each  $\Pi_i$  projects onto one column of the image. Each pixel of the cyclopean eye is originated from the back-projection ray  $\mathbf{d}_{i,r}$  (red). The  $\mathcal{N}_4$  neighbors of  $\mathbf{d}_{i,r}$  are  $\mathbf{d}_{i\pm 1,r}$  (blue) and  $\mathbf{d}_{i,r\pm 1}$  (green).

### 5.2. Initial plane hypotheses

As discussed in Section 3, each line cut is a possible location of intersection of a virtual cut plane with a planar surface in the scene. In order to propose an initial set of plane models  $\mathcal{P}_0$  for PEARL, we could generate all possible planes that can be obtained from two line cuts belonging to different planes  $\Pi$ , as originally proposed in [16, 28]. However and depending on the number of cut planes that are used, the set  $\mathcal{P}_0$  can easily become very large. We noticed that using only pairs of line cuts from neighboring cut planes  $\Pi_{i\pm(1,2)}$  drastically decreases the size of  $\mathcal{P}_0$  and is enough for initializing our piecewise-planar labeling approach. Since it is unlikely that line cuts intersecting different epipolar planes correspond to the same planar surface, we further reduce  $\mathcal{P}_0$  and only use pairs of line cuts that have a minimum of  $N_E$  epipolar lines of overlap ( $N_E = 10$  in this article).

### 5.3. Data and smoothness term

The data term  $D_{\mathbf{d}_{i,r}}$  for the back-projection ray  $\mathbf{d}_{i,r}$  is defined as

$$D_{\mathbf{d}_{i,r}}(f) = \begin{cases} \min(1 - E_i(r, x_f), \tau) & \text{if } f \in \mathcal{P}_0 \\ \tau & \text{if } f = f_\emptyset \end{cases}$$

where  $E_i$  is the joint energy associated with the virtual cut plane  $\Pi_i$ ,  $r$  is the row corresponding to the epipolar plane  $\Psi_r$  and  $\tau$

is a constant. The coordinate  $x_f$  is the column defined by the hypothesis  $f$ , corresponding to the intersection of  $\mathbf{d}_{i,r}$  with the plane indexed by  $f$ . Note that similarly to [10], the non-planar  $f_\emptyset$  label indicates that no satisfactory plane hypothesis can be assigned to  $\mathbf{d}_{i,r}$ . In this case, the back-projection ray  $\mathbf{d}_{i,r}$  has high probability of not intersecting the scene in a planar surface.

Inspired by the work of Sinha et al. [9], the smoothness term for neighboring nodes  $\mathbf{d}$  and  $\mathbf{e}$  is given by

$$V_{\mathbf{d}\mathbf{e}}(f_{\mathbf{d}}, f_{\mathbf{e}}) = \begin{cases} 0 & \text{if } f_{\mathbf{d}} = f_{\mathbf{e}} \\ \lambda_1 & \text{if } (\mathbf{d}, \mathbf{e}, f_{\mathbf{d}}, f_{\mathbf{e}}) \in S_1 \\ \lambda_2 & \text{if } (\mathbf{d}, \mathbf{e}, f_{\mathbf{d}}, f_{\mathbf{e}}) \in S_2 \\ \lambda_3 & \text{if } (\mathbf{d}, \mathbf{e}) \in S_3 \\ \lambda_4 & \text{if } (f_{\mathbf{d}} \vee f_{\mathbf{e}}) = f_\emptyset \\ 1 & \text{else} \end{cases} \quad (13)$$

where  $0 < \lambda_1 < \lambda_2 < \lambda_3 < 1$ , and the content of the sets  $S_1$ ,  $S_2$  and  $S_3$  is described next. Remark that no penalization is assigned to neighboring nodes receiving the same plane label, while in the case of one node obtaining the discard label  $f_\emptyset$ , a non-zero cost  $\lambda_4$  is added to the plane configuration  $\mathbf{f}$ .

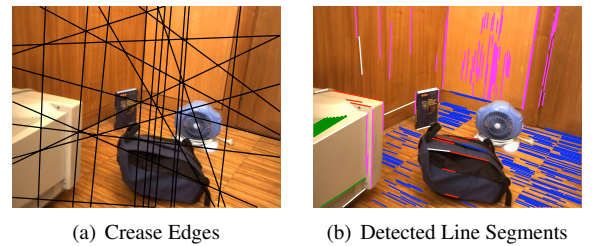


Figure 9: We show in (a) some crease edges obtained from intersections of two different planes in  $\mathcal{P}_0$ , while in (b) the result of the clustering of concurrent lines is shown. Each group of lines (different groups have different colors) provides a possible vanishing point location. The white line segments did not received any vanishing point label.

Following a similar reasoning as [9], plane transitions between neighboring nodes  $\mathbf{d}$  and  $\mathbf{e}$  are more likely to occur in the presence of crease or occlusion edges. A crease edge corresponds to the projection of the 3D line of intersection between two different planes in the scene, while occlusion boundaries arise from spatially separated objects in 3D whose image projections interfere with each other.

Let the point  $\mathbf{p}_{\mathbf{d},f_d}$  ( $\mathbf{p}_{\mathbf{e},f_e}$ ) be the projection of the intersection between the back-projection ray  $\mathbf{d}$  ( $\mathbf{e}$ ) and the plane associated to  $f_d$  ( $f_e$ ). In order to encourage plane label transitions at crease edges, we store in the set  $S_1$ , the quadruples  $(\mathbf{d}, \mathbf{e}, f_d, f_e)$  in which the points  $\mathbf{p}_{\mathbf{d},f_d}$  and  $\mathbf{p}_{\mathbf{e},f_e}$  are located on different sides of the crease edge defined by  $f_d$  and  $f_e$ . Whenever  $\mathbf{f}$  contains assignments located in  $S_1$ , then it incurs a penalization  $\lambda_1$  (Figure 9(a) shows some crease edges that are estimated from real imagery).

Occlusion edges usually coincide with visible 2D line segments in the input views and are often aligned with the vanishing directions of scene planes (Figure 9(b)). In order to find possible occlusion edges, we detect 2D line segments in the left view  $\mathbf{I}$  using the Line Segment Detector [29]. Each line segment is a possible location of an occlusion boundary. For clustering concurrent lines, we use the global vanishing point detection algorithm proposed by Antunes et. al [30]. The set  $S_2$  contains the quadruples  $(\mathbf{d}, \mathbf{e}, f_d, f_e)$  where the points  $\mathbf{p}_{\mathbf{d},f_d}$  and  $\mathbf{p}_{\mathbf{e},f_e}$  are located on different sides of a line segment that was clustered to a particular vanishing point, whose direction is orthogonal either to the planes associated to  $f_d$  or  $f_e$ . Finally,  $S_3$  contains the remaining pairs  $(\mathbf{d}, \mathbf{e})$  whose projections are on different sides of a line segment to which no vanishing point was assigned. Remark that in contrast to [9], we do not perform any line matching between the views, substantially decreasing the complexity of the algorithm.

#### 5.4. Plane refinement

The third step of PEARL (Section 2.2) is to re-estimate the plane model parameters using the inliers of the discrete labeling  $\mathbf{f}$ . Let  $\Omega_f$  be the plane associated to  $f$  to which has been assigned a non-empty set of inliers  $\mathbf{D}(f) = \{\mathbf{d} \in \mathcal{D} | f_{\mathbf{d}} = f\}$ . Each plane  $\Omega_f$  is refined by minimizing its plane parameters over the energies  $E$  via LM [25]:

$$\Omega_f^* = \min_{\Omega_f} \sum_{\mathbf{d}_{i,r} \in \mathbf{D}(f)} (1 - E_i(r, x_{\Omega})), \quad (14)$$

where  $x_{\Omega}$  is the column defined by the intersection of  $\mathbf{d}_{i,r}$  with  $\Omega$ . The new set of labels  $\mathcal{P}_1 = \{\Omega_f^*\}$  is then used in a new expand step, and we iterate between discrete labeling and plane refinement until the  $\alpha$ -expansion optimization does not decrease the energy of Equation 3 (which usually takes 2–3 iterations).

#### 5.5. Plane refinement after PEARL

We have discussed in Section 4.1 that SymStereo can be enhanced in case there is slant information available. The output of the global algorithm described previously, is the labeling  $\mathbf{f}$  that assigns to each back-projection ray  $\mathbf{d}$  a plane  $\Omega$ . The intersection of  $\mathbf{d}$  with  $\Omega$  defines a 3D point  $\mathbf{Q}$ , and  $\Omega$  also defines  $\alpha_1$  that is proportional to the 3D slant in the neighborhood of  $\mathbf{Q}$ . Following this, and as described in Section 4.1, the position  $\mathbf{Q}$  can be refined by iteratively optimizing  $\beta$ .

Let  $\Omega$  be the plane associated to label  $f$  to which has been assigned a non-empty set of inliers  $\mathbf{D}(f) =$

$\{\mathbf{d}_{i,r} \in \mathcal{D} | f_{\mathbf{d}_{i,r}} = f\}$ , and consider that  $\mathbf{Q}_{i,r}$  is the intersection between the ray  $\mathbf{d}_{i,r}$  and  $\Omega$  (refer to Figure 6). For each  $\mathbf{d}_{i,r}$ , we compute the corresponding *ideal*  $\beta_1$  and obtain a new back-projection ray  $\mathbf{d}_{i,r}^1$ . The new ray  $\mathbf{d}_{i,r}^1$  is located on the same epipolar plane, but on the virtual cut plane intersecting the point  $\mathbf{O}^1$  and the previously reconstructed point  $\mathbf{Q}_{i,r}$ . Given the new plane  $\Omega^1$ , a new homography mapping (see Equation 2) can be used for inducing improved symmetries, and from which the joint energy  $E_{i,r}^1$  is re-calculated. The new joint energies  $E_{i,r}^1$  are used in a new refinement step using LM (Section 5.4). We iterate between re-computing new back-projection rays  $\mathbf{d}_{i,r}^n$  and refining  $\Omega^n$  for a pre-defined number of times (4 in this article).

#### 5.6. Comparison between line cut reconstruction and semi-dense PPR

We show in Figure 10 a brief comparison between the line cut reconstruction algorithm described in Section 3 with the semi-dense PPR approach described in this section. In the case the virtual cut planes intersect planar surfaces with some texture and far from object discontinuities, the independent reconstruction along single virtual planes provides accurate results (example (a)). In scenarios containing multiple planes and complicated textures (examples (b-c)), the independent line cut reconstruction has some difficulties. These problems are solved using our semi-dense PPR pipeline that estimates planar surfaces in the scene along different virtual cut planes simultaneously and in a global manner.

### 6. Experiments in Piecewise-Planar Reconstruction

We proposed a new and original algorithm for detecting and estimating planar surfaces in the scene that combines SymStereo and PEARL optimization. For showing the effective advantages with respect to the existing PPR approaches, this section runs a set of experiments in PPR from a pair of stereo images, and compares the performance of the proposed algorithm with respect to three state-of-the-art approaches.

The evaluation is carried on a new dataset comprising challenging indoor and outdoor scenes (some examples are shown in Figures 11-13). The stereo pairs were acquired using a Bumblebee stereo camera from PointGrey, with a baseline of 24 cm and image resolution of  $1024 \times 768$  pixels. The scenes contain mostly planar surfaces, including a variety of complicated situations to traditional stereo methods e.g. low and/or repetitive textures, and surface slant.

#### 6.1. Compared Algorithms

The output of our algorithm (**SymS**) is a discrete set of plane hypotheses  $\mathcal{P}^{SymS}$  and a semi-dense 3D reconstruction. We compare these plane hypotheses with the ones obtained using two different approaches.

The first applies dense stereo (**DS**) for PPR and was proposed by Gallup et al. [10]. The authors start by obtaining a dense depth map with respect to the left view using local stereo. Then, plane hypotheses are generated using a sequential RANSAC procedure over the disparity map (refer to [10]

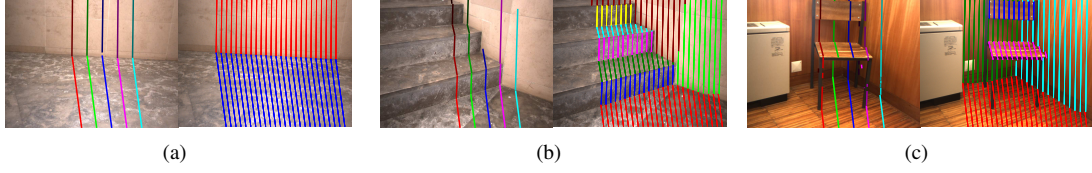


Figure 10: Comparison between independent line cut reconstruction and semi-dense PPR. For each example, we show the (independent) detection results along 5 virtual cut planes (left), and the final labeling results of the semi-dense PPR for 25 cut planes (right).

for details). Finally, a plane linking step is performed for combining near planes and/or single planes that are disjoint in the image. The output of this algorithm is the set  $\mathcal{P}^{DS}$  of plane hypotheses and a dense PPR.

The second approach that is compared was proposed by Sinha et al. [9], and is based on sparse stereo (SS). It detects and computes sparse correspondences, line segments and vanishing directions from the images. From these data, plane hypotheses are generated from specific histogram votings and RANSAC procedures. The output is the set  $\mathcal{P}^{SS}$  and a sparse PPR composed by 3D points and 3D line segments.

## 6.2. Accuracy analysis and parameter tuning

The objective is to compare the performance of DS, SymS and SS for generating plane hypotheses for the MRF plane labeling described in Section 2.3. Concerning the accuracy analysis, it is difficult to obtain the *ground truth* (GT) model parameters in each stereo pair of the dataset, which would involve a error prone and time consuming manual selection of point matches in the stereo views. We decided to use a different indicator for measuring the accuracy.

For each stereo pair, we manually define the planar region  $\mathcal{R}_k$  in the left view  $I$  that is associated to a particular plane  $\Omega_k$  in the scene (see Figure 11). Given the pixel-wise plane labeling  $\mathbf{f}$ , that was computed using the plane hypotheses generated from the algorithms described in Section 6.1, the accuracy of the estimation of  $\Omega_k$  is evaluated using the following metric:

$$P_k = \frac{\sum_{\mathbf{p} \in \mathcal{R}_k} \rho_{\mathbf{p}}(\mathbf{f}_{\mathbf{p}})}{\#\mathcal{R}_k}, \quad (15)$$

where  $\#\mathcal{R}_k$  is the number of pixels in the region. Remark that the accuracy analysis using  $P_k$  must be performed with caution. There is no guarantee that  $P_k < P_l$  means the plane  $\Omega_k$  was better estimated than  $\Omega_l$ . The proposed metric depends largely on the textures and illumination of the surfaces e.g. planar surfaces with low-texture and specularities will have a large  $P_k$  even tough the corresponding plane model is well estimated. On the contrary, we are in the opinion that the metric  $P_k$  is adequate for comparing different estimations of the same plane  $\Omega_k$ .

Assume that we use two different algorithms for obtaining two different sets of plane hypotheses, say  $\mathcal{P}^{A1}$  and  $\mathcal{P}^{A2}$ , which are used as input to the global plane labeling described in Section 2.3. After the graph-cut optimization, we have the assignments  $\mathbf{f}^{A1}$  from  $\mathcal{P}^{A1}$  and  $\mathbf{f}^{A2}$  from  $\mathcal{P}^{A2}$  for each image pixel. Following this, we can compute for each GT plane  $\Omega_k$  the

photo-consistency metrics  $P_k^{A1}$  and  $P_k^{A2}$ . In case  $P_k^{A1} < P_k^{A2}$ , then the first algorithm generated a plane hypothesis that better fits the input images, which most probably means that  $\Omega_k^{A1}$  is more accurate than  $\Omega_k^{A2}$ . We noticed in practice that this empirical comparison is a very good accuracy indicator in real-world scenarios.

The parameters that are used in the different algorithms were manually tuned using the GT labeling on a subset of stereo pairs of the dataset, whose results are not shown in the experimental comparison. These values are kept constant for all the remaining experiments. Concerning our SymS algorithm, we decided to use  $M = 25$  virtual cut planes for the best compromise between accuracy and runtime. Concerning the MRF labeling (see Section 2.3), the parameters are constant and the same for all three plane hypotheses generators, namely  $\rho_{max} = 0.8$ ,  $\gamma = 0.6$ ,  $m = 1$  and  $M = 2$ .

## 6.3. Comparison results

The dense PPR results obtained using DS, SymS and SS as plane hypotheses generators for the pixel-wise plane labeling are shown in Figure 11.

In the first two examples, the scene is composed by two and three planes, respectively, which are mostly fronto-parallel to the cameras. In these cases, the three algorithms work well and provide approximately similar results. SS has some problems distinguishing the vertical planes in the example (b), which is mainly due to lack of features in the wall on the right. Both examples shown in the second row contain, besides other planes, a highly slanted surface (blue in example (c) and green in example (d)). Our algorithm is able to detect and accurately reconstruct this surfaces, whereas DS and SS clearly have difficulties handling this amount of slant. The examples (e) and (f) show scenes containing many planes at different distances from the camera. SymS is able to detect all the planes and provides the most accurate plane hypotheses, being less sensitive to the surface-camera distance when compared to DS and SS.

The last row shows two examples containing scene with difficult textures and illumination conditions. SS is not able to provide acceptable plane hypotheses for the MRF labeling so that no plane assignment is obtained. DS is still able to cope with the complicated texture of example (g), but completely fails in the example (h), where the joint effect of high slant and complicated textures are major challenges for dense stereo matching. Our approach recovers all the planes, and can even distinguish the close planes of example (g) corresponding to the floor and the carpet.



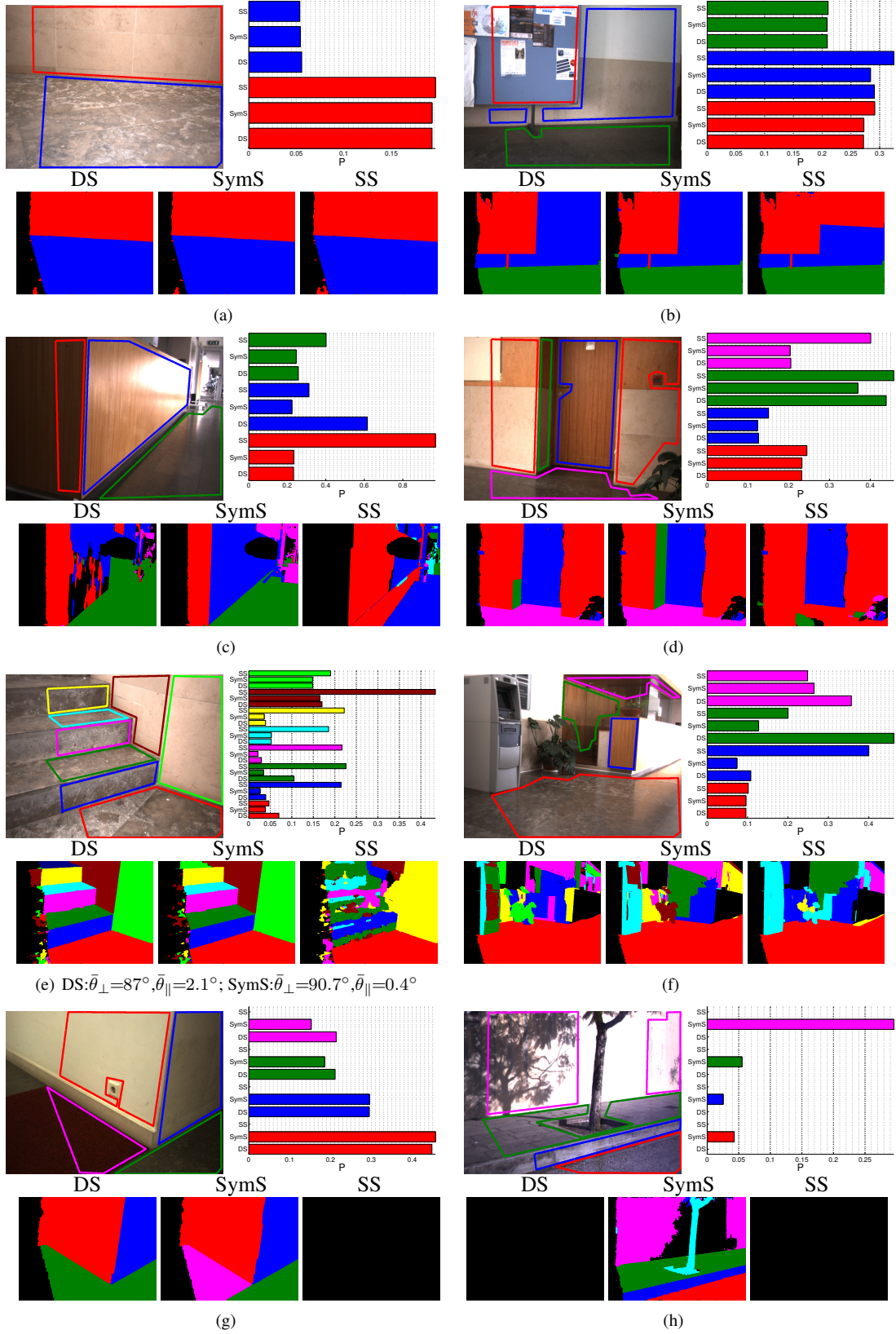


Figure 11: Comparison between DS, SymS and SS for PPR. For each example we show (top, left)  $I$  with GT labeling, different colors correspond to different planes; (top, right) mean photo-consistency  $P$  in the GT region for each algorithm, each color identifies a particular plane; and (bottom) pixel-wise plane assignment obtained using the different algorithms as plane hypotheses generators, different colors identify different planes. The black label refers to the discard label  $f_0$ . As additional accuracy indicator, we manually identified for the example (e) the planes that are mutually orthogonal (e.g. blue and red) and parallel (e.g. green and red). We present the mean angles  $\bar{\theta}_\perp$  and  $\bar{\theta}_\parallel$  between the perpendicular and the parallel planes, respectively.

Finally and for the sake of completeness, the run-times (without the final MRF labeling) for each algorithm in the images shown in Figure 11 are: 1–2 min for SymS (the runtime mostly depends on the number of line cuts that are estimated (Section 3)), 2 min for DS, and approximately 3 min for SS. These are straight-forward and unoptimized implementations in Matlab, except for  $\alpha$ -expansion optimization, for which the public available code of [31, 32, 33, 34] in C++ is used.

#### 6.4. Two view piecewise models

As discussed in [35], the depth error in stereo vision is related with the correspondence error by a multiplication factor known as the geometric resolution that depends on the baseline and on the focal length. We assume that the maximum allowed relative depth error should be 2%. From our evaluation, this value is reached, for the case of our algorithm and in the images shown in Figure 11, for a depth of around 12 m. This will be our depth reconstruction limit, so that we will not reconstruct surfaces further away from this bound.

Figure 12 and Figure 13 show plane labeling and 3D reconstruction results in indoor and outdoor scenes, respectively. This is the type of environments targeted by the PPR algorithms described in [14, 8, 9, 10]. While these methods require multiple views, our approach is able to reach competitive results using only a stereo pair. The labeling results are exclusively based on photo-consistency and proximity, which explains the poorly defined region borders in some examples. Such issue can be easily solved using a more sophisticated pixel-wise plane labeling MRF, similar to the one used in Section 5 that incorporates crease and occlusion edge information. We chose not to do so in order to better assess the accuracy of our plane pose estimation.

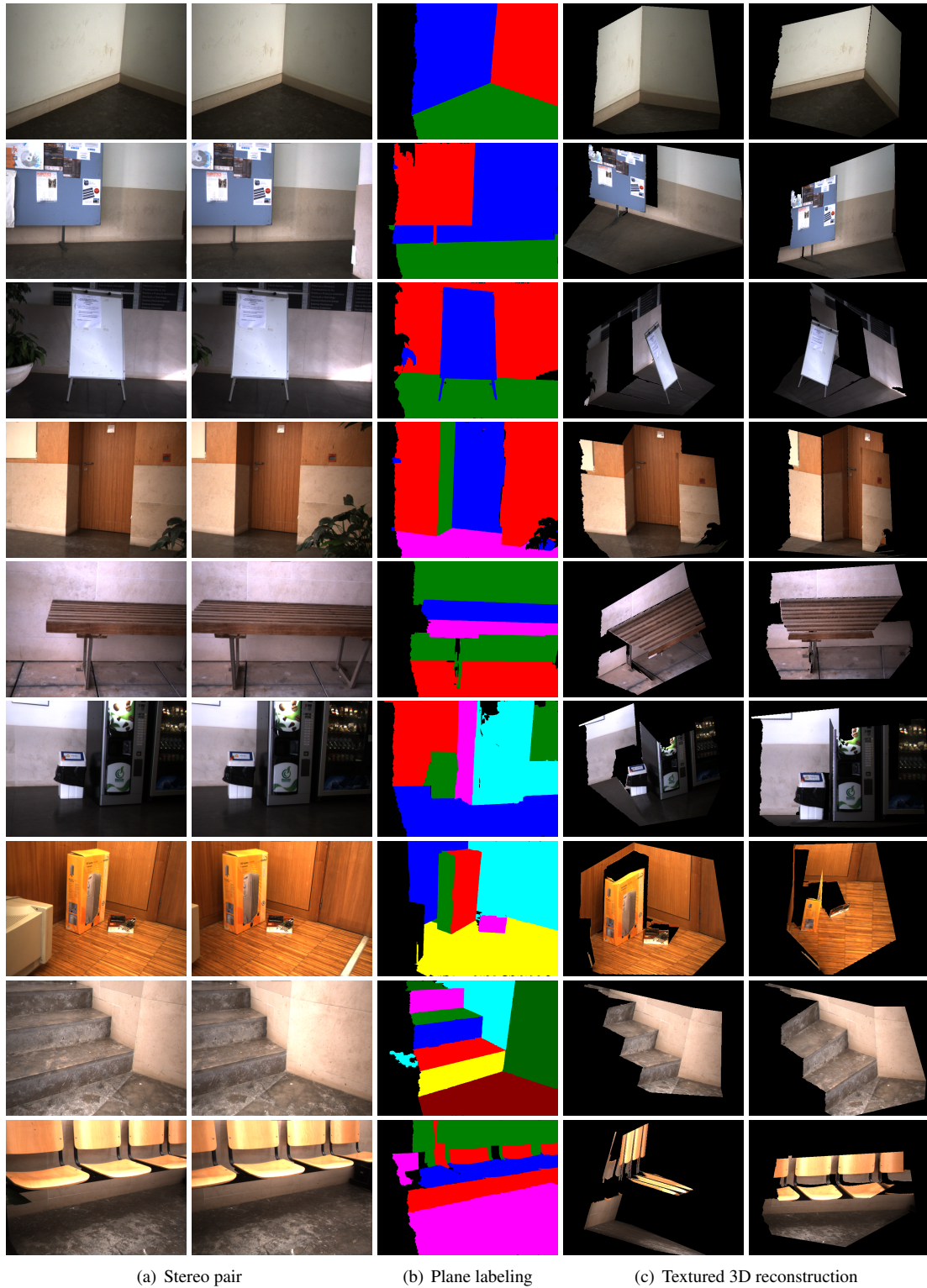
## 7. Conclusion

The paper presents an automatic piecewise planar reconstruction algorithm from two views. Unlike other existing approaches, the stereo depth estimation and the detection of planar surfaces are accomplished in a tight and coupled manner by combining SymStereo with PEARL [2]. This enables to take full advantage of the strong planarity prior, with the algorithm being able to accurately segment and reconstruct the planes contained in the scene. The effectiveness of the scheme is proved by comparison with two different state-of-the-art approaches in several challenging indoor and outdoor scenarios.

As a final comment, it can be claimed that the energy-based model fitting can either be applied to dense stereo reconstruction or to a sparse point-cloud model. The former would substantially increase the computational complexity without bringing obvious benefits, while the latter would avoid the use of the smoothness term for regularizing the PEARL energy minimization. Thus, the symmetry-based semi-dense stereo provides a trade-off between the two, playing a key role in the success of the overall approach.

## References

- [1] M. Antunes, J. P. Barreto, Symstereo: Stereo matching using induced symmetry, in: Under review in IJCV, 2013.
- [2] H. Isack, Y. Boykov, Energy-based geometric multi-model fitting, IJCV.
- [3] D. Scharstein, R. Szeliski, R. Zabih, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, IJCV.
- [4] S. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: CVPR, 2006.
- [5] R. Klette, N. Kruger, T. Vaudrey, K. Pauwels, M. van Hulle, S. Morales, F. Kandil, R. Haeusler, N. Pugeault, C. Rabe, M. Lappe, Performance of correspondence algorithms in vision-based driver assistance using an on-line image sequence database, Vehicular Technology, IEEE Transactions on.
- [6] T. Werner, A. Zisserman, New techniques for automated architectural reconstruction from photographs, in: ECCV, 2002.
- [7] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénus, R. Yang, G. Welch, H. Towles, Detailed real-time urban 3d reconstruction from video, IJCV.
- [8] Y. Furukawa, B. Curless, S. Seitz, R. Szeliski, Manhattan-world stereo, in: CVPR, 2009.
- [9] S. Sinha, D. Steedly, R. Szeliski, Piecewise planar stereo for image-based rendering, in: ICCV, 2009.
- [10] D. Gallup, J.-M. Frahm, M. Pollefeys, Piecewise planar and non-planar stereo for urban scene reconstruction, CVPR.
- [11] Y. Zhang, M. Gong, Y.-H. Yang, Local stereo matching with 3d adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy, in: ICPR, 2008.
- [12] M. Bleyer, C. Rhemann, C. Rother, Patchmatch stereo - stereo matching with slanted support windows, in: BMVC, 2011.
- [13] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM.
- [14] A. Bartoli, A random sampling strategy for piecewise planar scene segmentation, CVIU.
- [15] M. Antunes, J. P. Barreto, C. Premebida, U. Nunes, Can stereo vision replace a laser-rangefinder?, IROS.
- [16] M. Antunes, J. P. Barreto, X. Zabulis, Plane surface detection and reconstruction using induced stereo symmetry, in: BMVC, 2011.
- [17] Y. Furukawa, J. Ponce, Accurate, dense, and robust multiview stereopsis, PAMI.
- [18] A. Klaus, M. Sormann, K. Karner, Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure, in: ICPR, 2006.
- [19] M. H. Lin, C. Tomasi, Surfaces with occlusions from layered stereo, PAMI.
- [20] M. Bleyer, M. Gelautz, A layered stereo algorithm using image segmentation and global visibility constraints, in: ICIP, 2004.
- [21] M. Bleyer, C. Rother, P. Kohli, Surface stereo with soft segmentation, in: CVPR, 2010.
- [22] S. Baker, R. Szeliski, P. Anandan, A layered approach to stereo reconstruction, in: CVPR, 1998.
- [23] S. Birchfield, C. Tomasi, Multiway cut for stereo and motion with slanted surfaces, ICCV.
- [24] H. Tao, H. S. Sawhney, R. Kumar, A global matching framework for stereo computation, ICCV.
- [25] R. I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.
- [26] R. T. Collins, A space-sweep approach to true multi-image matching, in: CVPR, 1996.
- [27] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, M. Pollefeys, Real-time plane-sweeping stereo with multiple sweeping directions, in: CVPR, 2007.
- [28] M. Antunes, J. P. Barreto, Semi-dense piecewise planar stereo reconstruction using symstereo and pearl, in: 3DimPVT, 2012.
- [29] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, G. Randall, LSD: A Fast Line Segment Detector with a False Detection Control, PAMI.
- [30] M. Antunes, J. P. Barreto, A global approach for the detection of vanishing points and mutually orthogonal vanishing directions, in: CVPR, 2013.



(a) Stereo pair

(b) Plane labeling

(c) Textured 3D reconstruction

Figure 12: Indoor results produced by our PPR algorithm.

- [31] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, PAMI.
- [32] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts, PAMI.
- [33] Y. Boykov, V. Kolmogorov, PAMI.
- [34] A. Delong, A. Osokin, H. N. Isack, Y. Boykov, Fast approximate energy

- minimization with label costs, IJCV.
- [35] D. Gallup, J.-M. Frahm, P. Mordohai, M. Pollefeys, Variable base-line/resolution stereo, in: CVPR, 2008.



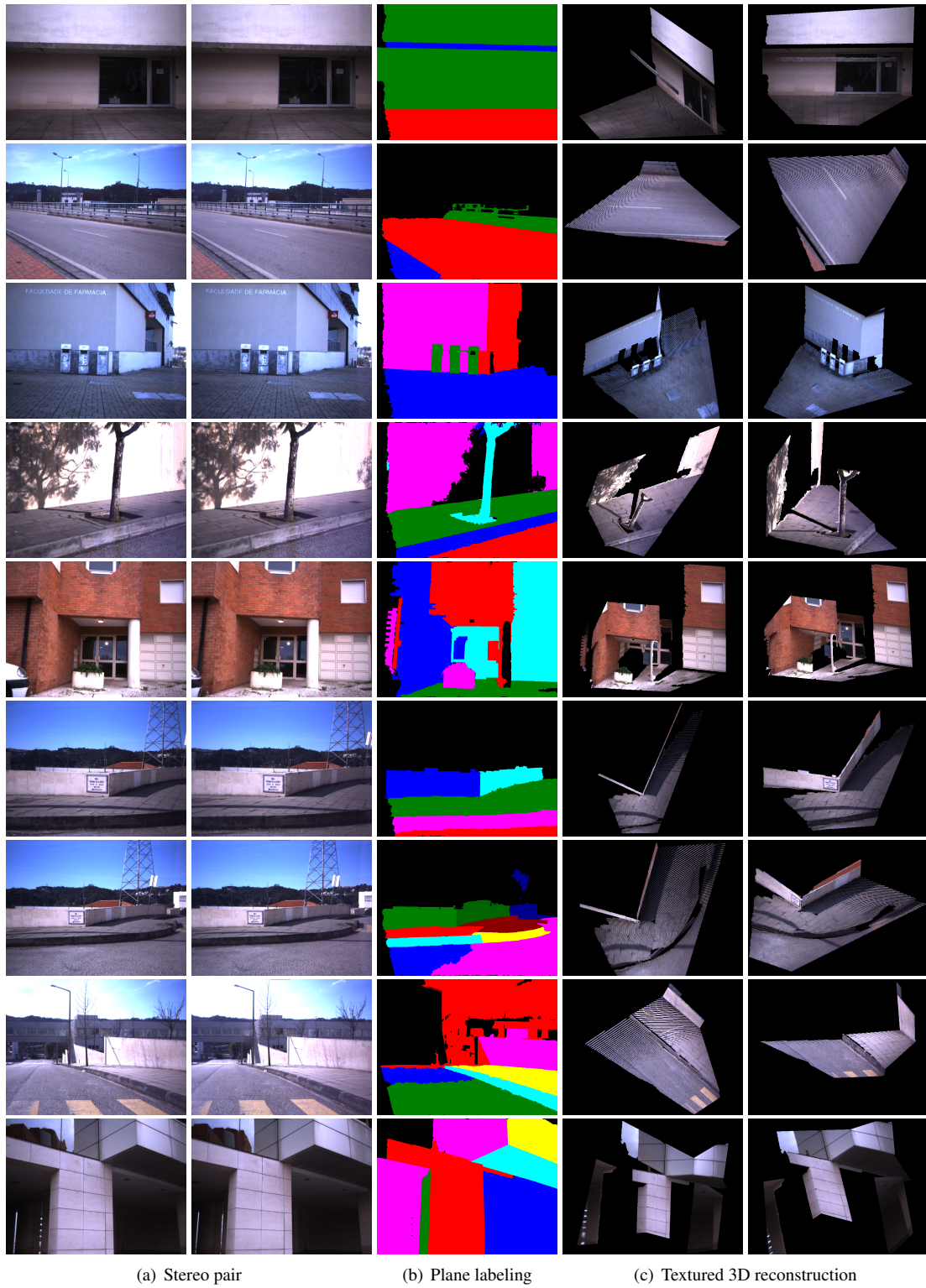


Figure 13: Outdoor results produced by our PPR algorithm.